

A Hybrid Algorithm for the Equal Districting Problem

Yunfeng Kong ¹[0000-0002-0777-3116]

¹ Key Laboratory of Geospatial Technology for the Middle and Lower Yellow River Regions, Ministry of Education, Henan University, 475000, Kaifeng, China
yfkong@henu.edu.cn

Abstract. The equal districting problem (EDP) arises in applications such as political redistricting, police patrol area delineation, sales territory design and some service area design. The important criteria for these problems are district equality, contiguity and compactness. A mixed integer linear programming (MILP) model and a hybrid algorithm are proposed for the EDP. The hybrid algorithm is designed by extending iterative local search (ILS) algorithm with three schemes: population-based ILS, variable neighborhood descent (VND) local search, and set partitioning. The performance of the algorithm was tested on five areas. Experimentation showed that the instances could be solved effectively and efficiently.

Keywords: Equal districting problem; Mixed integer linear programming; Hybrid algorithm.

1 Introduction

Districting problems have been widely applied in geography, economics, environmental science, politics, business, public service and many other areas. The equal districting problem (EDP) arises in applications such as political redistricting, police patrol area delineation, sales territory design and some service area design. The basic criteria for the problem are balance, compactness and contiguity [1] [2].

There are two general approaches to solve the EDP: exact methods and heuristics [2] [3]. It is hard to solve the EDP by exact methods due to its computational complexity [4]. Salazaraguilar et al. (2011) proposed a bi-objective programming model and solved an instance with 150 units and six territories in four hours [5]. Duque et al. (2011) have shown that only very small instances of the p-regions problem with 49 units could be solved [6]. A new model proposed by Plane et al. (2019) was used to solve political districting instances with 25~36 units in 6~53,632 seconds [7]. In practice, some districting criteria were not fully considered in model formulations [3].

Early studies had also modelled the problem as a capacitated p-median facility location problem, and developed several location-based solution approaches [8][9][10][11]. It is not necessary to determine district centers in solving the districting problem; however, districting measures such as contiguity and compactness rely on district centers [2]. Kong et al. (2019) proposed a center-based model for solving the districting problem, in which the district centers were identified by a weighted K-medoids algorithm [12].

Three classes of heuristic methods were developed for districting problems: the local-search based heuristics such as greedy search, simulated annealing, tabu search, and old bachelor acceptance search [1][13][14][15][16][17]; evolutionary methods such as evolutionary algorithm, genetic algorithm, and scatter search[18][19][20][21]; and swarm intelligence methods such as artificial bee colony[22]. However, the high-performance methods are always time consuming. For example, the PEAR algorithm [23] needs 3 hours to solve an instance on a super computer with 131K processors.

This article aims to propose a high-performance hybrid algorithm for the EDP. The hybrid algorithm is designed by extending iterative local search (ILS) algorithm with three schemes: population-based ILS, variable neighborhood descent (VND) local search, and set partitioning. The performance of the algorithm was tested on five areas. Experimentation showed that the instances could be solved effectively and efficiently.

2 Mathematical formulation

For a geographic area, let set $V=\{1, 2 \dots n\}$ denote n basic areal units. Each unit i has an attribute value p_i . Let c_{ij} indicate whether units i and j share a border and N_i be a set of units that are adjacent to unit i ($N_i = \{j \mid c_{ij}=1\}$). Let variables d_{ij} be the distance between units i and j . The districting problem in this article is to partition set V into K equal, compact and contiguous districts. The EDP is to delineate a geographic area into K equal, compact and contiguous districts. A mixed integer linear programming (MILP) model is formulated as follows.

$$\text{Min.} \sum_{i \in V} \sum_{k \in V} p_i d_{ik} x_{ik} / \sum_{i \in V} p_i \quad (1)$$

$$\text{S.T.} \sum_{k \in V} x_{ik} = 1, \forall i \in V \quad (2)$$

$$(1 - \varepsilon) \bar{Q} y_k \leq \sum_{i \in V} p_i x_{ik} \leq (1 + \varepsilon) \bar{Q} y_k, \forall k \in V \quad (3)$$

$$\sum_{k \in V} y_k = K \quad (4)$$

$$f_{ijk} \leq (n - K) x_{ik}, \forall i \in V, j \in N_i, k \in V \quad (5)$$

$$f_{ijk} \leq (n - K) x_{jk}, \forall i \in V, j \in N_i, k \in V \quad (6)$$

$$\sum_{j \in N_i} f_{ijk} - \sum_{j \in N_i} f_{jik} \geq x_{ik}, \forall i \in V, k \in V, i \neq k \quad (7)$$

$$x_{ik} = \{0, 1\}, \forall i \in V, k \in V \quad (8)$$

$$y_k = \{0, 1\}, \forall k \in V \quad (9)$$

$$f_{ijk} \geq 0, \forall i \in V, j \in N_i, k \in V \quad (10)$$

The objective function (1) is to minimize the average travel distance from basic units to their district centers. Constraints (2) ensure that each unit i is assigned to only one district. Constraints (3) confirm that districts are almost equal with a predefined error ε , e.g. $\varepsilon=5\%$. Note that $\bar{Q} = \sum_{i \in V} p_i / K$. Constraint (4) requests that the area is divided into K regions. The district contiguity is ensured by constraints (5), (6) and (7). It is a variant of the network-flow based model [6] to guarantee the district contiguity. Constraints (5) and (6) ensure that a flow may only be passed through neighbor units in the same district. If unit i does not serve as the center unit, constraints (7) state that one unit of flow must be created from this unit. Constraints (8), (9) and (10) define the decision variables.

3 Algorithm design

A hybrid algorithm for the EDP was proposed based on iterative local search (ILS) algorithm. ILS is a simple, easy to implement, and quite effective metaheuristic for discrete optimization problems. It starts from an initial solution and iteratively improves the solution by local search and perturbation. In this article, the standard ILS algorithm was enhanced by three schemes: population-based search, variable

neighborhood descent (VND) search, and set partitioning. Given parameters such as population size ($psize$), perturbation strength ($strength$), number of consecutive loops that the best solution is not updated ($mloops$), time limit for set partitioning ($tlimit$), the hybrid algorithm is outlined as follows:

```

1.  $P = \text{GenerateInitialSolutions}(psize);$ 
2.  $pool = \text{null};$ 
3.  $s_{best} = \text{Best}(P);$ 
4.  $notImpr = 0;$ 
5. While  $notImpr < mloops$ :
6.   Select a solution  $s$  from  $P$  randomly;
7.    $s' = \text{Perturbation}(s, strength);$ 
8.    $s'' = \text{VNDsearch}(s');$ 
9.    $s^* = \text{updateCenters}(s'')$ 
10.  If  $f(s^*) < f(s_{best})$ :  $s_{best} = s^*, notImpr = 0;$ 
11.  else:  $notImpr += 1;$ 
12.   $pool = \text{UpdateAreaPool}(pool, s^*);$ 
13.   $P = \text{UpdatePopulation}(P, s^*);$ 
14.   $s = \text{SetPartitioning}(pool, t);$ 
15. Output  $s$ .
```

The hybrid algorithm maintains a number of candidate solutions. In step (1), initial solutions for the EDP are constructed by a weighted K-medoids algorithm [12]. In step (6), a solution is randomly selected from the population as the incumbent solution. After solution perturbation and local search, in step (13), the population is updated according to the solution objectives and the similarities among the solutions. The population-based extension of ILS may enhance the diversification of local search. The elite and diverse solutions are selected by the algorithm.

The design of local search operators in optimization algorithm is critical for repeatedly improving the incumbent solution. The one-unit shift is a widely used operator that moves a boundary unit to its neighboring district while maintaining the connectivity of its original district. Butsch et al. introduced three operators for districting problems: shift, double shift, and swap [24]. In the proposed algorithm, two local search operators are used to improve the solutions. Instead of simple local search in ILS, VND search is used in step (8). VND is repeatedly exploring different neighborhood structures by the change of neighborhoods until the solution cannot be improvement. Using VND search, ILS algorithm can easily find solutions in local optimum.

Perturbation is one of the key components of ILS algorithm. Using local search operators, the solution may easily reach to local optima. A ruin and recreate procedure is useful to perturb the EDP solution from local optima. There are multiple methods to ruin the solution such as deleting some boundary units, deleting all the units in some districts, and deleting some units in a connective region. The ruined solution must be repaired to be a feasible solution.

In step (9), the district centers are updated by selecting the best unit in each district.

In step (14), the best solution might be improved by a set partitioning procedure. In each ILS loop, all the districts in solution s^* are recorded in the list $pool$. Each district record includes the units in the district and its objective value. At the end of ILS, thousands or more districts will be recorded in the pool. A set partitioning problem (SPP) model could be used to select a better solution from the candidate service areas. Let Ω be the set of districts identified in step (9), U_i and O_i be the set of basic units in each

district i and its objective, respectively. A SPP model is used to select K districts from Ω . Let Ω_j be a subset of Ω , $\Omega_j = \{i | i \in \Omega, j \in U_i\}$, the SPP model is formulated as follows:

$$\text{Min. } \sum_{i \in \Omega} O_i x_i \quad (11)$$

$$\text{S.T. } \sum_{i \in \Omega_j} x_i = 1, \forall j \in V \quad (12)$$

$$\sum_{i \in \Omega} x_i = K \quad (13)$$

$$x_i = \{0,1\}, \forall i \in \Omega \quad (14)$$

4 Experimentation

Five geographic areas (Fig. 1) are selected to test the EDP. The main attributes of the areas are summarized in Table 1.

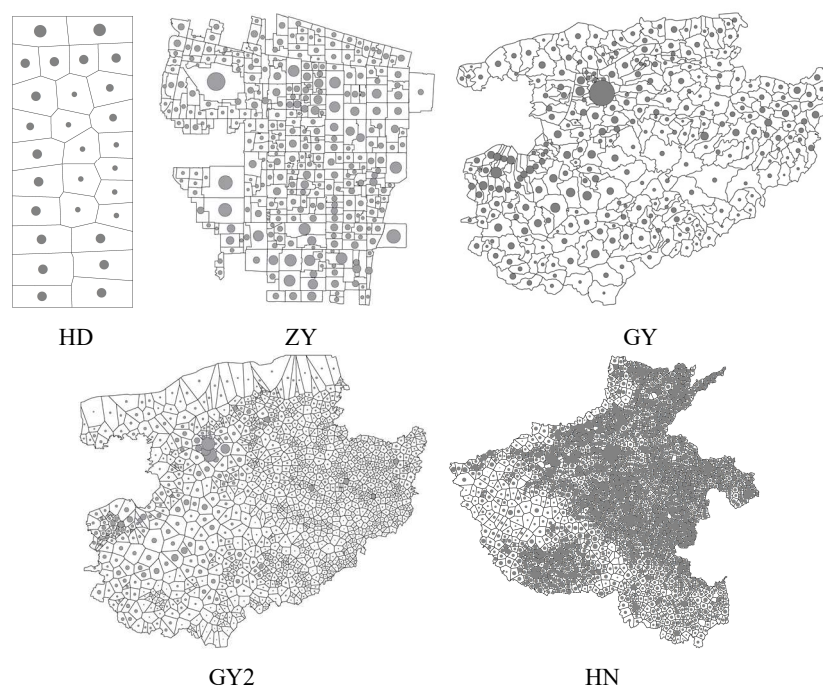


Fig. 1. The study areas

Table 1. Main attributes of the study areas

Area	HD	GY	ZY	GY2	HN
Geography	Res. community	County	Urban	County	Province
No. of basic units	28	297	324	1,276	2,145
Sum of attribute value	2,420	36,824	3,873	819,812	96,918,036

The proposed algorithm was implemented by using the Python programming language, and was executed in PyPy 7.0, a fast and compliant implementation of the Python language (www.pypy.org). The variables p_i , c_{ij} and d_{ij} for each area were prepared in ArcGIS 10 (www.esri.com). All the software ran on a desktop computer with Intel Core I7-6700 CPU 3.40-GHz, 8 GB RAM and the Windows 10 operating system.

The problem instances were solved by exact and heuristic methods, respectively. First, each instance model was solve by CPLEX Optimizer 12.6. A solution with lower bound (LB), upper bound

(UB), gap between LB and UB (MIPGap) and computation time (Time) could be obtained from CPLEX optimizer. Note that only the instances on small and medium areas HD, GY and ZY can be solved by CPLEX. Second, each instance was repeatedly solved by the hybrid algorithm for ten times. The parameters were set as follows: $\varepsilon=5\%$, $psize=10$, $strength=5\%$, $mloops=100$, and $tlimit=100$ second. The heuristic solutions are evaluated in terms of the average gap to lower bound (Gap), the standard deviation among 10 solutions (Dev) and the average computation time (Time).

Districting results from areas HD, GY and ZY are summarized in Table 2. The solution results show that all the instances can be optimally solved by CPLEX optimizer; and the solutions from the hybrid algorithm approximate to the optimal solutions or the lower bounds with small gaps ranging from 0.00% to 1.19%. Part of solutions are illustrated in Fig. 2, 3 and 4.

Table 2. Districting results from the areas HD, GY and ZY

Area	K	ε	CPLEX				Hybrid		
			LB	UB	MIPGap	Time/s	Gap	Dev	Time/s
HD	2	5%	0.099	0.099	Opt.	0.99	0.00%	0.00%	0.32
HD	3	5%	0.084	0.084	Opt.	1.20	0.00%	0.00%	0.44
HD	4	5%	0.071	0.071	Opt.	1.14	0.00%	0.00%	0.52
HD	5	5%	0.060	0.060	Opt.	1.31	0.00%	0.00%	0.62
GY	2	5%	8.149	8.149	Opt.	6,098.57	0.00%	0.00%	6.20
GY	3	5%	6.085	6.085	Opt.	3,990.47	0.04%	0.06%	6.18
GY	4	5%	5.135	5.135	Opt.	11,145.66	1.19%	0.03%	7.51
GY	5	5%	4.616	4.616	Opt.	7,588.53	0.16%	0.07%	9.92
GY	6	5%	4.185	4.185	Opt.	11,303.26	0.24%	0.37%	6.67
GY	7	5%	3.751	3.751	Opt.	3,576.36	0.00%	0.00%	6.17
GY	8	5%	3.529	3.529	Opt.	5,390.92	0.22%	0.09%	6.92
GY	10	5%	3.120	3.120	Opt.	4,823.14	0.00%	0.01%	7.18
ZY	2	5%	0.979	0.979	Opt.	3,480.72	0.00%	0.00%	5.24
ZY	3	5%	0.831	0.831	Opt.	3,796.54	0.00%	0.00%	5.75
ZY	4	5%	0.717	0.717	Opt.	4,292.85	0.01%	0.04%	5.84
ZY	5	5%	0.624	0.624	Opt.	4,758.05	0.05%	0.03%	5.80
ZY	6	5%	0.562	0.562	Opt.	5,811.00	0.08%	0.02%	6.04
ZY	7	5%	0.513	0.513	Opt.	5,927.70	0.01%	0.01%	6.00
ZY	8	5%	0.477	0.477	Opt.	4,544.64	0.02%	0.04%	7.32
ZY	10	5%	0.425	0.425	Opt.	4,973.77	0.02%	0.01%	7.03
ZY	15	5%	0.351	0.351	Opt.	4,973.77	0.17%	0.10%	9.84
ZY	20	5%	0.297	0.297	Opt.	5,397.47	0.29%	0.19%	15.54

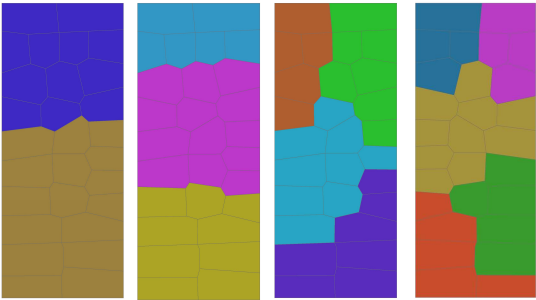


Fig. 2. Districting maps of the area HD

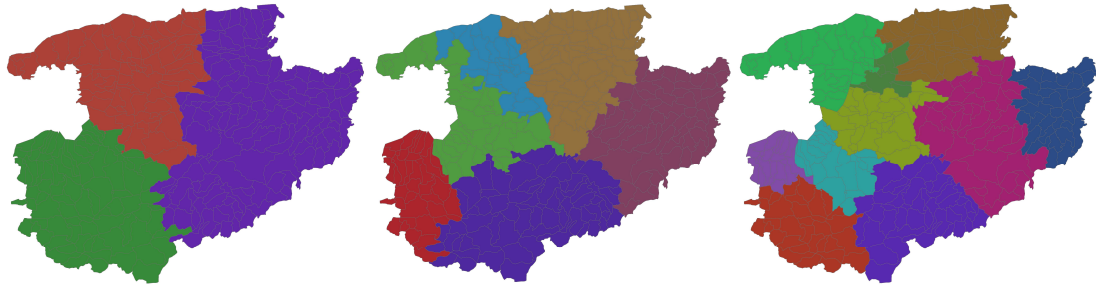


Fig. 3. Districting maps of the area GY

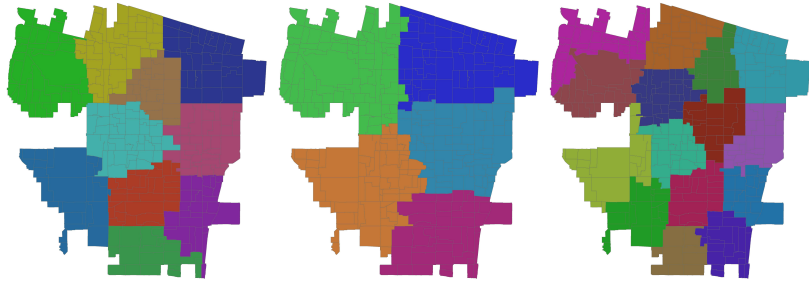


Fig. 4. Districting maps of the area ZY

Districting results from areas GY2 and HN are summarized in Table 3. Due to large numbers of basic units in the two areas, the CPLEX failed to solve the instance models with memory overflow errors. Among 10 solutions for each instance, the best, average and worst objective are shown in columns Min, Avg and Max, respectively. The standard deviation of objectives and the average computing time are illustrated in columns Dev and Time, respectively. The results show that the large problems could be efficiently solved in 2~9 minutes with very small deviations ($<0.18\%$). Part of districting maps are illustrated in Fig. 5 and 6.

Table 3. Districting results from areas GY2 and HN

Area	K	ε	Min	Avg	Max	Dev	Time/s
GY2	5	5%	4.439	4.442	4.443	0.03%	115.78
GY2	10	5%	3.108	3.109	3.110	0.03%	123.96
GY2	15	5%	2.544	2.548	2.551	0.08%	268.82
GY2	20	5%	2.174	2.177	2.181	0.11%	165.05
HN	5	5%	69.762	69.766	69.769	0.00%	302.46
HN	10	5%	46.548	46.560	46.586	0.02%	330.24
HN	15	5%	37.562	37.571	37.581	0.02%	546.56
HN	20	5%	31.781	31.795	31.810	0.03%	404.85
HN	25	5%	28.291	28.307	28.327	0.04%	539.99
HN	30	5%	25.854	25.864	25.879	0.03%	340.52
HN	35	5%	23.764	23.798	23.832	0.09%	323.62
HN	40	5%	22.244	22.296	22.333	0.13%	498.67
HN	45	5%	20.760	20.802	20.860	0.18%	388.39
HN	50	5%	19.541	19.590	19.612	0.12%	410.77

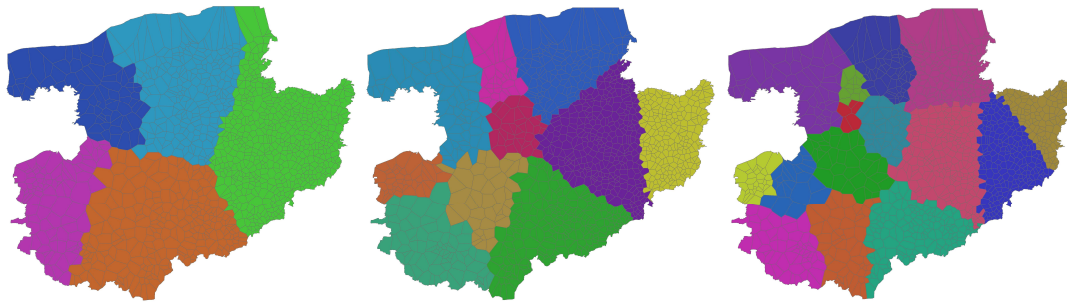


Fig. 5. Districting maps of area GY2



Fig. 6. Districting maps of area HN

5 Conclusion

A mathematical model and a hybrid algorithm were proposed for the EDP in this paper. The MILP model is different from existing models. First, it is a center-based model, and thus it is easy to formulate an objective function on district compactness. Second, the constraints on district contiguity is simplified by the center units of districts. Third, the EDP is defined as an equally capacitated p -median problem with constraints of district contiguity. As a result of the center-based formulation, the proposed model could optimally solve EDP instances with 324 basic units, while the districting models in [6] and [7] could only solve small instances with 16-49 or 25-36 units, respectively.

The proposed hybrid algorithm is different from existing algorithms in some aspects. The perturbation in ILS algorithm plays an important role in escaping from local optima and an appropriate perturbation strength could guide the ILS search approaching the global optimum. The algorithm was designed based on the standard ILS algorithm, and further enhanced by four schemes: VND local search, population-based ILS, center updating, and set partitioning. The experimentation showed that the EDP instances could be solved effectively and efficiently. The solutions found by the hybrid algorithm approximate optimal solutions or the lower bounds with very small gaps, and are also robust with deviations less than 0.20%.

References

1. Ricca, F., Simeone, B.: Local search algorithms for political districting. *European Journal of Operational Research*, 2008, 189(3): 1409-1426.
2. Kalcics, J. Districting Problems. In: Laporte, G., Nickel, S., da Gama, F. S.(eds) *Location Science*, Springer, 2015: 595-622.
3. Ricca, F., Scozzari A., Simeone B., et al.: Political districting: from classical models to recent approaches. *A Quarterly Journal of Operations Research*, 2011, 9(3): 223-254.
4. Keane, M. C.: The size of the region-building problem. *Environment and Planning A*, 1975, 7(5): 575-577.
5. Salazaraguilar, M. A., Riosmercado, R. Z., Gonzalezverlarde, J. L.: A bi-objective programming model for designing compact and balanced territories in commercial districting. *Transportation Research Part C: Emerging Technologies*, 2011, 19(5), 885-895.
6. Duque, J. C., Church, R. L., Middleton, R. S., et al.: The p -Regions problem. *Geographical Analysis*, 2011, 43(1): 104-126.
7. Plane, D. A., Tong, D. and Lei, T.: Inter-person Separation: A New Objective Standard for Evaluating the Spatial Fairness of Political Redistricting Plans. *Geographical Analysis*, 2019, 51: 251-279.

8. Hess, S. W., Weaver, J. B., Siegfeldt, H. J., et al. Nonpartisan Political Redistricting by Computer. *Operations Research*, 1965, 13(6): 998-1006.
9. Hojati, M.: Optimal political districting. *Computers & Operations Research*, 1996, 23(12): 1147-1161.
10. George, J. A., Lamar, B. W., Wallace, C. A.: Political district determination using large-scale network optimization. *Socio-Economic Planning Sciences*, 1997, 31(1):11-28.
11. Nemoto, T., Hotta, K.: Modelling and solution of the problem of optimal electoral districting. *Communications of Operations Research of Japan*, 2003, 48(4): 300–306. (in Japanese)
12. Kong, Y., Zhu, Y., Wang, Y.: A center-based modeling approach to solve the districting problem, *International Journal of Geographical Information Science*, 2019, 33(2), 368-384.
13. Garfinkel, R., Nemhauser, G. L.: Optimal Political Districting by Implicit Enumeration Techniques. *Management Science*, 1970, 16(8): 495-508.
14. Nygreen, B.: European assembly constituencies for Wales - comparing of methods for solving a political districting problem. *Mathematical Programming*, 1988, 42(1): 159–169.
15. Mehrotra, A., Johnson, E. L., Nemhauser, G. L.: An Optimization Based Heuristic for Political Districting. *Management Science*, 1998, 44(8):1100-1114.
16. Damico, S. J., Wang, S., Batta, R., et al. A simulated annealing approach to police district design. *Computers & Operations Research*, 2002, 29(6): 667-684.
17. Bozkaya, B., Erkut, E., Laporte, G., et al.: A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*, 2003, 144(1): 12-26.
18. Wei, B. C., Chai, W. Y.: A multiobjective hybrid metaheuristic approach for gis-based spatial zoning model. *Journal of Mathematical Modelling & Algorithms*, 2004, 3(3):245-261.
19. Bacao F, Lobo V, Painho M, et al. Applying genetic algorithms to zone design. *Soft Computing*, 2005, 9(5): 341-348.
20. Chou, C., Chu, Y., Li, S., et al.: Evolutionary strategy for political districting problem using genetic algorithm. *International Conference on Conceptual Structures*, 2007.
21. Salazar-Aguilar, M. A., Rios-Mercado, R. Z., Gonzalez-Velarde, J. L., et al.: Multiobjective scatter search for a commercial territory design problem. *Annals of Operations Research*, 2012, 199(1): 343-360.
22. Rincón-García, E. A., Gutiérrez-Andrade, M. Á., De-Los-Cobos-Silva, S. G., et al.: ABC, A Viable Algorithm for the Political Districting problem. In: *Scientific Methods for the Treatment of Uncertainty in Social Sciences*. Springer International Publishing, 2015:269-278.
23. Liu, Y., Choc, W. K., Wang, S.: PEAR: a massively parallel evolutionary computation approach for political redistricting optimization and analysis. *Swarm and Evolutionary Computation*, 2016, 30: 78–92.
24. Butsch, A., Kalcsics, J., Laporte, G., 2014. Districting for Arc Routing. *Inform Journal on Computing*, 26(4), 809-824.

Foundation Support: The National Natural Science Foundation of China, No. 41871307.